**CS349 NETWORKS LAB ASSIGNMENT – 2**

**(NPTEL/Coursera video lectures)**

[**https://drive.google.com/open?id=1KdiAY-myGlFOoeJC1bSEdAMDqdS2dqO\_**](https://drive.google.com/open?id=1KdiAY-myGlFOoeJC1bSEdAMDqdS2dqO_)

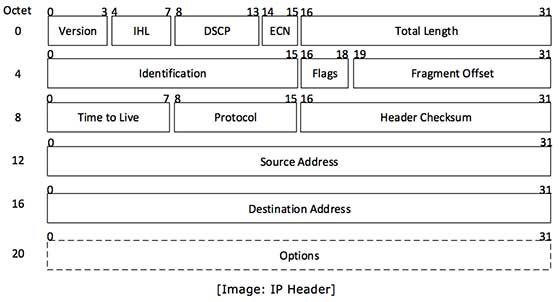
NAME – SUNNY KUMAR ROLL NO -170101068

**Question 1.** Various Protocols used by the application at different layers are:

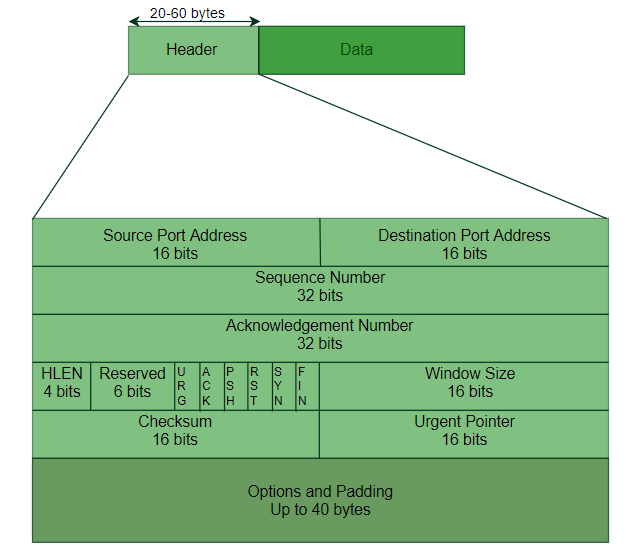
* **Physical Layer: Ethernet II**: The first 6 bytes is for the destination address, the next 6 bytes is for source address, next 2 bytes for ether type: IPv4 or IPv6, 46 to 1500 bytes for data, generally consisting upper layer headers. The last 4 bytes is for Frame Check Sequence or CRC.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dest. MAC | Source MAC | Ether Type | Data | FCS |
| 6 bytes | 6 bytes | 2 bytes | 46 to 1500 bytes | 4 bytes |

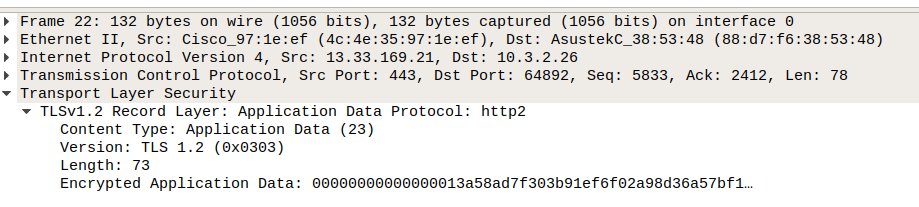
* **Network Layer: IPv4**: **Version** of the IP protocol (4 bits), which is 4 for IPv4, **IHL** (Header Length), **DSCP** (Differentiated Services Code Point), **ECN** (Explicit Congestion Notification) allows end to end notification of network congestion without dropping packets, **Total length** is 16-bit field which defines the entire packet size, **identification** is unique Packet Id for identifying the group of fragments of a single IP datagram (16 bits), **Flags:** 3 flags of 1 bit each: reserved bit, do not fragment flag, more fragments flag. **Fragment Offset** represents the no. of data bytes ahead of the particular fragment in the particular Datagram. **Time to Live** represents Datagram’s lifetime, **Protocol** field is for the name of protocol, **Header Checksum** for checking errors in the datagram header. **Source Address** sender IP address and **Destination Address** receiver IP address.



* **Transport Layer: TCP**: **Source Port** (16 bits) holds the port address of the application that is sending the data segment. **Destination Port** (16 bits) holds the port address of the application in the host that is receiving the data segment. **Sequence number** (32 bits) holds the sequence number. **Acknowledgement number** (32 bits) holds the acknowledgement number. **Header length (HLEN)** is a 4-bit field that indicates the length of the TCP header by no. of 4-byte words in the header. **Flags** are 6 1-bit control bits that control connection establishment, termination, abortion, flow control, mode of transfer etc. **Window Size** tells window size of the sending TCP. **Checksum** field holds the checksum for error control. **Urgent Pointer** is used to point to data that is urgently required.

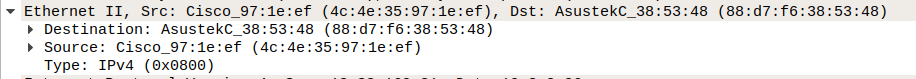


* **TLSv1.2 Record Layer: Application Data Protocol**: This layer acts as an intermediate between transport layer and application layer. Its packet format: **Content Type**, **Version**, **Length** and **Encrypted Application Data**.

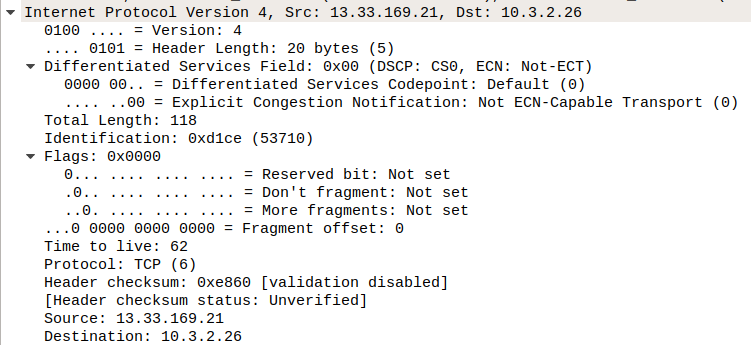


**Question 2.**

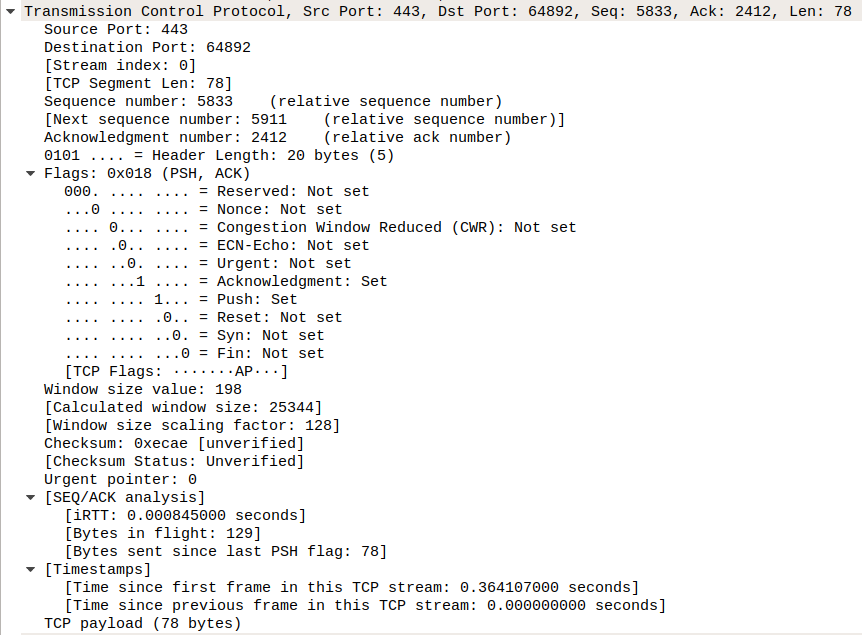
* **Ethernet II:** **Src: ASUSTekc\_38:53:48 (88:d7:f6:38:53:48)**: Hardware MAC address of my PC(source). **Dst: Cisco\_97:1e:ef (4c:4e:35:97:1e:ef)**: Hardware MAC address of destination. **Type**: IPv4 (internet protocol version type).



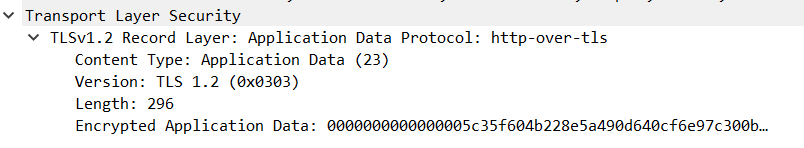
* **Internet Protocol Version 4:** **Src: 13.33.169.21**- This is source IP address. **Dst: 10.3.2.26**- destination IP address. **Version** is **4** as we are using IPv4. **Header Length** is (5) **20 bytes** as it counts in 4 Bytes word. **Differentiated Services Field**: **0x00** indicates particular quality of service needs from the network. **Total Length** is **118**: is the length of datagram (entire packet size). **Identification**- unique packet id (**0xd1ce**). **Flags: 0x0000** i.e. first 3 bits used for flag bits and rest 13 bits are **fragment offset**. Here **000** is 3-bit flag in which **Bit 0**: reserved, must be zero, **Bit 1** is **0** that denotes **Don’t Fragment** and **Bit 2** is 0. **Time to Live (62)**- maximum time the datagram is allowed to remain in internet system. **Protocol** is TCP and **Header Checksum:** status unverified (validation disabled).



* **Transmission Control Protocol:** **Src Port: 443**: source port, which is sending packets. **Dst Port: 64892**: This is receiving/Destination port which is receiving packets. **Sequence number** and **Acknowledgment number** is 5833/2412 (relatively). **Flags**:**018** used to indicate a particular state of connection. **Window size value**: **198**: This is the space for the incoming data. Checksum is **0xecae.** **Urgent Pointer: 0:** If the URG flag is set, then this field is an offset from the sequence number indicating the last urgent data type.



* **TLSv1.2 Protocol:** Since the http is encrypted by TLS, it cannot be decoded by Wireshark separately. But it shows Application Data Protocol that has field of **content type:** Application Data, **Version** is **TLS 1.2** and **Length** is 296 bytes that is entire packet size.



**Question 3.** Protocols used by the application to achieve the important functionalities like ‘play’, ‘pause’ and ‘download’ are **HTTP**, **TCP**, **IPv4**, **TLSv1.2** and **Ethernet II**.

**HTTP:**

* HTTP follows a classical client-server model, with a client opening a connection to make a request, then waiting until it receives a response.
* TCP is in charge of setting up a reliable connection between two machines and HTTP uses this connection to transfer data between the servers and client.

**TCP:**

* It is evident from the traces that TCP layer handles all the handshaking which is done for the establishment and termination of the connection.
* TCP provides host to host connectivity and control. If the packet is lost, it requests for retransmission.
* It is optimized for accurate delivery.

**IPv4:**

* This version of IP is used as the basis of the internet, and it establishes all the rules and regulations for the computer networks that function on the principle of packet exchange.
* The good thing about the IP address is that it is used as a unique identifier for computing devices that are connected to a local network or internet.

**TLSv1.2:**

* It provides privacy and data integrity between two communicating devices.
* It provides authentication and data encryption between servers and machines operating over a network.

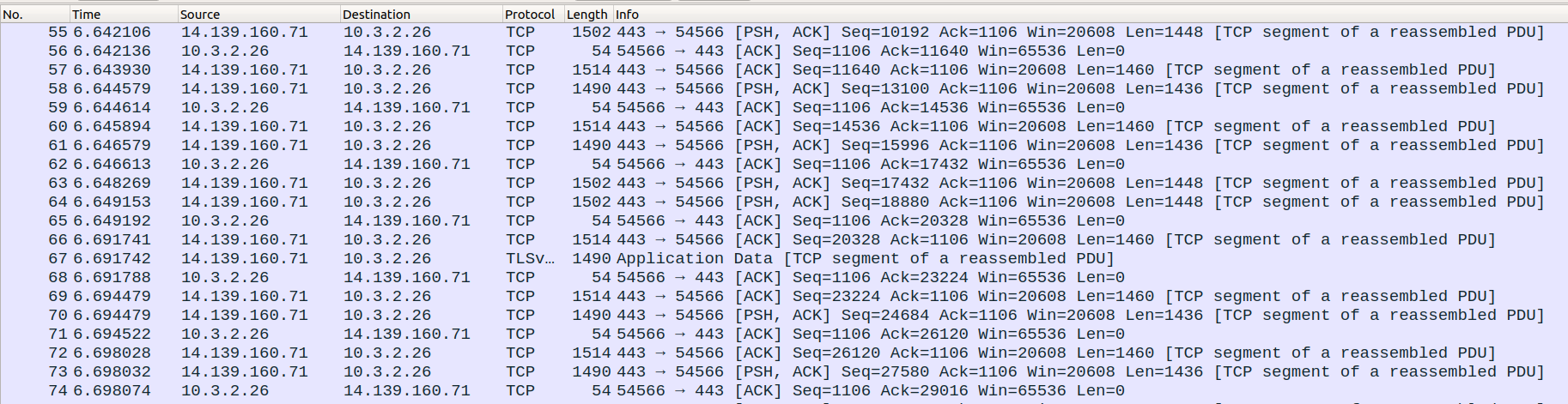
**Ethernet II:**

* Ethernet lying in data link layer contains information about MAC addresses of source and destination.
* It determines which switching protocols are used that is important for segmenting the data stream and preventing data congestion.

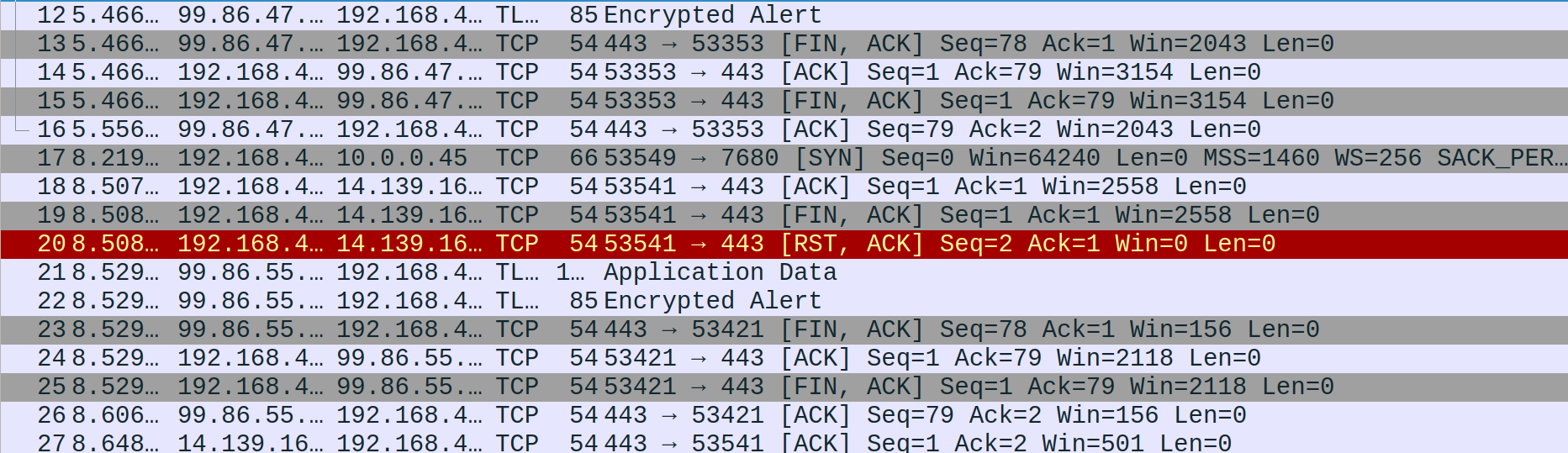
**Question 4**. I am mentioning here 3 functionalities of the application (**nptel.ac.in**): **Download the video**, **Pause the running video** and **running video**.

* **Download the video:** During downloading video, following traces displayed.

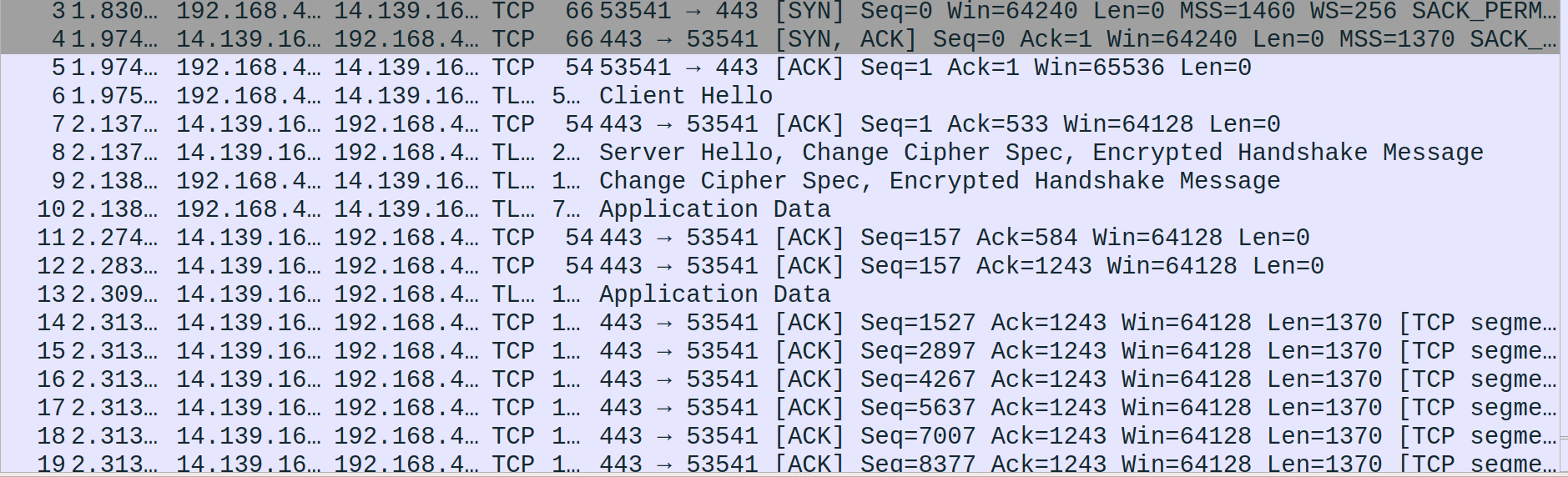
Initially connection is established through 3-way handshake and followed by TLS handshake. TCP segments continuously transferred from server to our computer. PSH flags received during downloading the video ensures data should be sent immediately.



* **Pause the running video:** After the video was paused, FIN flags received from server **99.86.47.53** which stands **Finished** means there is no more data in the sender. It terminates all TCP connections and so data transfer is stopped.



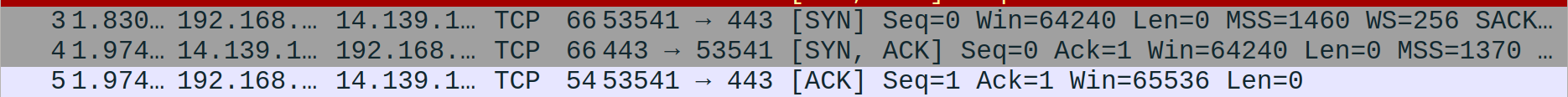
* **Running video:** First a sequence of TCP connection establishment through 3-way handshake. This got response from TCP after a TLS handshake to ensure security. While video is running it gives response through continuously getting TCP data transfer and obtaining **Application Data**.



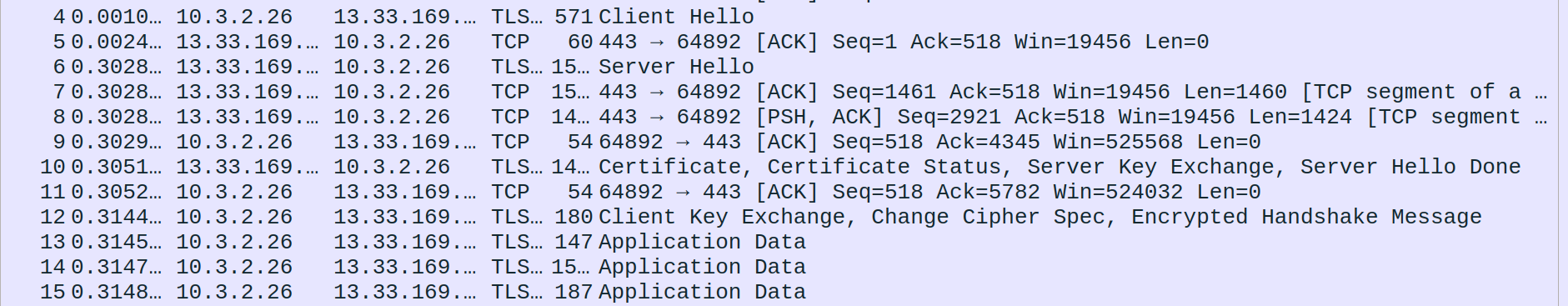
The following sequence of message exchanges are observed:

* **TCP Establishment:** Before a client attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections, this is called passive open. Once the passive open is established, a client may initiate an active open. Establishing a normal TCP connection requires these steps:

1. The client sends a [SYN] frame for requesting the server to synchronize.
2. The server sends the [SYN, ACK] frame for acknowledging the request for synchronization that was sent by the client.
3. The client then sends [ACK] acknowledging the request from the server.



* **TLS Handshaking Message:** This protocol is used to exchange all the information required by both sides for the exchange of the actual application data by TLS and to negotiate the **secure attributes of a session**. The TLS Handshake protocol allows authenticated communication between the server and client. This protocol allows the client and server to agree upon encryption keys before the selected application protocol begins to send data. There are various messages as **Client Hello, Server Hello, Change Cipher spec, Encrypted Handshake Message.**



**Question 5.** Statistical Analysis:

|  |  |  |  |
| --- | --- | --- | --- |
| Details | Morning (10:38 AM) | Afternoon (12:12 PM) | Night (11:39 PM) |
| Address A | 10.3.2.26 | 192.168.43.128 | 10.3.2.26 |
| Address B | 13.33.169.21 | 13.35.130.79 | 13.33.169.115 |
| Throughput (bits/s) | 130k | 3.138k | 128k |
| Average RTT (ms) | 4.07 | 62.01 | 6.92 |
| Packet Size (bytes) | 385 | 349 | 364 |
| No. of packets lost | 0 | 0 | 0 |
| TCP packets | 27 | 34 | 29 |
| UDP packets | 0 | 0 | 0 |
| Response per Request | 1.25 | 1 | 1.23 |

**Question 6**. Yes, I have captured packets at 3 different hours as mentioned in the above table and got multiple sources from which content is being sent and their IPs are:

|  |  |  |  |
| --- | --- | --- | --- |
| Time | 10:38 AM | 12:12 PM | 11:39 PM |
| IP | 13.33.169.21 | 13.35.130.79 | 13.33.169.115 |

Here, I have used my room IP in morning and night and Jio-wifi in afternoon for experiment.

**Reasons for multiple IPs**

* To prevent traffic from being exchanged via the gateway, speeding things up and reducing the load.
* To compensate for a host that’s down at that moment
* To serve web pages based on the user location. While using ping with domain name or type the domain name in a browser, the IP address is fetched from a DNS server. So, the application uses whatever IP address it gets from the DNS server (probably using round-robin).